

本周对 Twitter 数据的分析作了一些实验，同时也进一步学习了自然语言处理的基本方法。在实验过程中主要应用 NLTK 的 Python 包去实现话题检测与跟踪工作。主要过程可以描述如下：

- 1、分词(Tokenize)，NLTK 提供了专门的分词函数 `sent_tokenize()` 对文档中的句子做拆分，以及 `word_tokenize()` 函数从句子中拆分单词。
- 2、对句子中的词按词性标注，这里主要是按照英语的语法规则将句子中的每个单词加上连词 (CC)、介词 (IN)、名词 (NN) 等标注，函数为 `pos_tag()` 为下一步抽取命名实体做准备。标注后的句子类似下面：

```
... The/AT grand/JJ jury/NN commented/VBD on/IN a/AT number/NN of/IN
... other/AP topics/NNS ,/, AMONG/IN them/PPO the/AT Atlanta/NP and/CC
... Fulton/NP-tl County/NN-tl purchasing/VBG departments/NNS which/WDT it/PP
... said/VBD ``/`` ARE/BER well/QL operated/VBN and/CC follow/VB generally/R
... accepted/VBN practices/NNS which/WDT inure/VB to/IN the/AT best/JJT
... interest/NN of/IN both/ABX governments/NNS "/" ./.

```

- 3、抽取命名实体 (NE, Named Entity)，命名实体的抽取需要按照指定的语法规则 (relation) 找出所有的名词，如 `<DT|PP|$>?<JJ>*<NN>`，该正则表达式描述了典型的主谓宾结构，最后返回其中的 NN 即为命名实体。
- 4、构建 NE 共生图 (Co-occurrence Graph)，凡是在同一条 Tweet 中出现的 NE 相互之间添加一条边，边的权值为两个 NE 共同出现的次数。共生图表示为无向图 $G=<V, E>$ ，V 以数组表示，而 E 则用三元组数组表示，三元组为 `<NO1, NO2, Weight>`，其中 NOX 为节点在 V 数组中的序号，Weight 为权值。
- 5、计算节点 Betweenness 中心度，对共生图进行分割，不同的子图构成话题，记录话题中的 NE 以及话题对应的 tweet，将其存储到数据库。
- 6、以上过程以天为时间窗分析 Twitter 数据。相邻两天的话题按照相同 NE 的数量计算相似度，如果两天的话题相似度超过 40% 则将这两个话题合并。

上述过程已经进行到第 5 步，由于后面的两天去上课所以没有继续下去。

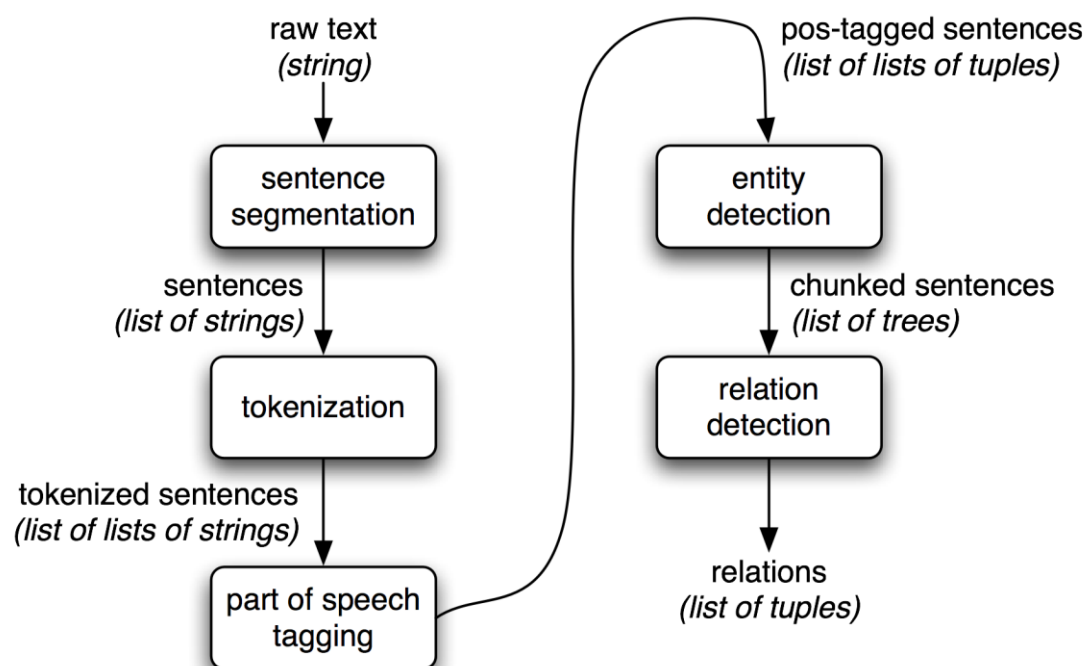


Figure 1 NLTK Named Entity Extraction Architecture

Label 与 Annotation 问题

这是在阅读Landesberger对大图可视化的综述时看到的一个较新的研究方向。作者讨论认为如何将大图中的节点以及节点相应的label有效地在一个图中表示,是未来值得研究的一个方向。在网上搜索文献,发现去年和今年真有论文做这种标注问题,如去年VAST的论文

“Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations”是IBM公司对数据进行实时点标识,今年CHI上的论文“Contextifier: automatic generation of annotated stock visualizations”实现了在线新闻标注工具,将公司股票交易线图中关键走势与该公司的新闻关联起来,以标注形式表示,如图2。

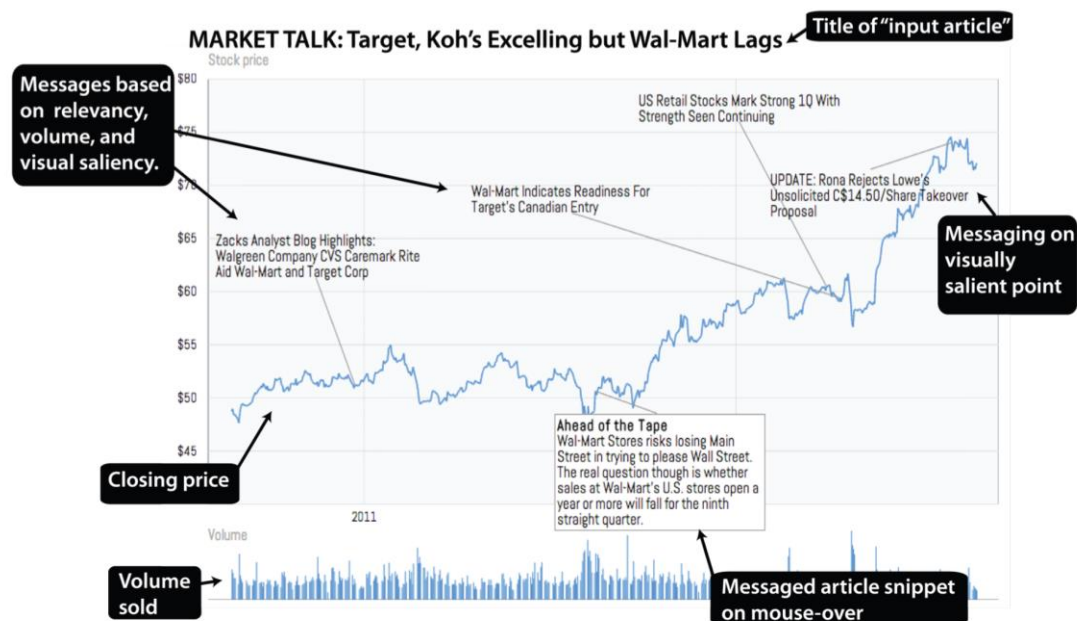


Figure 2 Contextifier 实现的股票交易标注

标注本质上可以定义为一类优化问题:

- Λ_f 是图形特征 (Feature) 集合 F 中元素 f 可以标注的位置集合
- Λ 是 F 中所有特征的可标注位置集合
- 定义函数 $\lambda: F \rightarrow \Lambda$, 将特征 f 映射到 Λ 中的某个位置上, 即 $\lambda(f) = \lambda_f \in \Lambda_f$ 。
- 定义代价函数 $COST: \Lambda \rightarrow \mathbb{N}$, 为选择 λ 所获得的代价

标注问题就是求最小代价。

Labeling Problem

Instance: Let F be a set of graphical features to be labeled.

Question: Find a label assignment that minimizes the following function:

$$\sum_{i \in F} \sum_{j \in \Lambda_i} COST(\lambda(i)) P(i, j)$$

Where:

$$P(i, j) = \begin{cases} 1, & \text{if } \lambda(i) = j, \\ 0, & \text{otherwise} \end{cases}$$

and

$$\sum_{i \in F} \sum_{j \in \Lambda_i} P(i, j) = |F|$$

Where:

$$\sum_{j \in \Lambda_i} P(i, j) = 1, \quad i \in F.$$

对于标注问题，可以分为：点标注、线标注以及面标注。

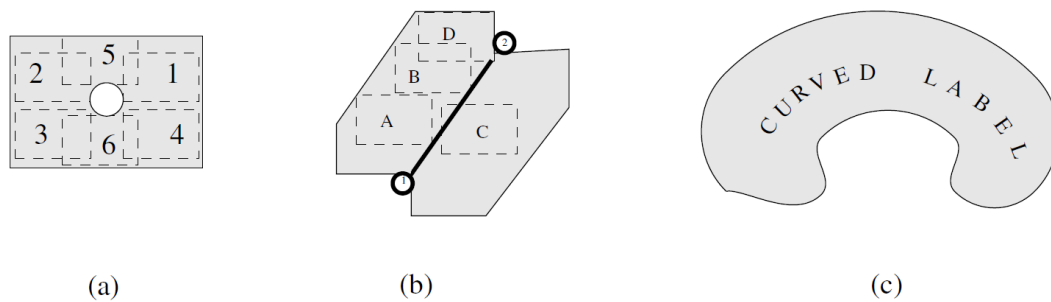


Figure 3 标注的三种形式

根据标注形式的不同，代价函数的计算方法也有所不同，有人证明即使是相当简单的点标注也是 NP 问题。最近一些年有一些算法去近似求解此类问题，根据实际应用限制一些条件居多。

我的想法是做一个类似于图 3 中 c 的面标注。如果应用到 ThemeRiver (Stack Graph) 中可以认为是将每个话题的文字按照话题曲线的流向排列，而不是简单的横向放置。昨天做了简单的设计，可以设计代价函数为话题所占据的面积，根据面积大小将文字设置在其中并将文字拆分成单个字符（个数为 n ）分别对应放置到曲线均匀切分的 n 个位置上。

产生这样的想法我觉得是我对信息可视理解的进一步提升。以前总是从可能遇到的实际问题出发思考可视化如何去帮助解决问题，所以即使最后做出了一个应用，也只是简单地照搬可视化中现有的结果，没有新意，也很难创新。因此，一度我对可视化产生了怀疑，觉得可视化就是简单地使用常规的柱状图、环图、树图去解决问题，真正的可视化做什么呢，跟图形学有什么关系？在产生这样的想法后，对可视化的理解比以前更深入了。